Terminal and Command-Line Cheat Sheet

simon

<2018-03-08 Thu>

Contents

Open a Terminal and you can directly type the stuff below.

If you have time, enjoy this essay by the science-fiction author Neal Stephenson.
Editing file is most of the job, so please also consider using a descent editor.
The author uses and recommends GNU Emacs, because it simply rocks!

---

1 Getting Help

```
man <command>
```

Quit by pressing `q`.

```
man ls
man cd
man mkdir
```

2 The `TAB` key

Whenever entering (long) paths or file names, the `TAB` key comes in very handy, because it **autocompletes** the end or proposes how to complete. **Autocompletion** is so handy...

Imagine you want to enter in this fictional directory, by typing all these components:

```
cd /data/home/alturi/project/long-filename.ext
```

Prone error !! Instead, the `TAB` key is magic, try:

```
cd /d[TAB]ata/h[TAB]ome/al[TAB]turi/pro[TAB]ject/lo[TAB]ng-filename.ext
```

[0]This document was generated the: Wednesday 21st March, 2018, 15:42

When you type ambiguous character (e.g., `pro` should point to your fictional folder `project/` or `product/`), the completion does not work. In that case, hit `TAB` twice to view all the possible matches and then type a few more characters.

3 History of the command line

Just use `ARROW UP` and `DOWN` to navigate through the history.

List all the recent history:

```
history
```

4 Where I am

```
pwd
```

Show the absolute path.

5 Create new directory

```
mkdir <name>
```

You can also create the directory and couple of subfolders:

```
mkdir -p my-project/this/that
```

6 Change directory

```
cd <directory>
```

For example, go to the previous created folder, and verify you are in:

```
cd my-project/this/that
pwd
```

Go at one level up (parent directory) and verify again:

```
cd ..
pwd
```

Go to the folder `that/` then go at two levels up:

```
cd that/
pwd
cd ../..
pwd
```

Note that:

```
cd
```

go to the `$HOME` folder.

7 List the content of a directory

```
ls <directory>
```

and without any `<directory>` name, list the current folder.

List all the files, even the hidden ones:

```
ls -a
```

List the files and sort them by reverse order of modified time:

```
ls -rt1
```

List the files with some useful information (permissions, owner, size etc.)

```
ls -l
```

List recursively through the subfolders:

```
ls -R
```

8 Read the content of a file

```
less <filename>
```

Quit with `:q`.

9 Display the first `N` lines (last `N` lines)

```
head -nN <filename>
tail -nN <filename>
```

For example, display the first 5 commands:

```
head -n5 ~/.bash_history
```

10 Clear the terminal window (just cosmetic)

```
clear
```

Nothing is erased, it is pure cosmetic by refreshing.

11 Copy file / directory

```
cp <source> <target>
```

For example, copy the history of the command lines and list the folder:

```
cp ~/.bash_history ~/my-history
ls -rt1
```

After creating a new folder, copy the file into it:

```
cp my-history my-project/this/that
ls my[TAB]-project/[TAB]this/[TAB]that/
```

Copy folders:

```
cp -R my-project my-project2
ls -R my-project2
```

12 Rename file / directory

```
mv <source> <target>
```

13 Remove file / directory

```
rm <filename>
rm -fr <filename>
```

The option `-f` means force. Be careful !!

14 Search files

```
find <dir> -name "<filename>" -type f
```

For example, list all the files with the extensions `.fastq.gz` in the current folder:

```
find . -name "*.fastq.gz" -type f -print
```

Find all Pearl files `.pl` containing the occurence *xls* and print the line:

```
find . -type f -name "*.pl" -print | xargs grep -nH xls
```

15 Copy files / directory through the network

```
rsync -av --progress <source> <target>
```

For example, push local folder to server `toto.tata.univ-paris-diderot.fr`:

```
rsync -av --progress my-project username@toto.tata.univ-paris-diderot.fr:~/
```

Pull remote folder:

```
rsync -av --progress username@toto.tata.univ-paris-diderot.fr:~/my-project my-project2
```

Be careful with the trailing slash `/`. Explanations later !

16 Check what is going on

```
htop
```

17 Kill active process

```
CONTROL c
```

Or you can find the process number with:

```
ps -fe | less
```

and identify the guilty.

18 Disconnect the session

```
CONTROL d
```